

(For new non-provisional applications under 37 CFR § 1.53(b))

§ § § § §

1

If any fees are inadvertently omitted or if any additional fees are required or have been overpaid, please appropriately charge or credit those fees to Conley, Rose & Tayon, P.C. Deposit Account Number 03-2769/5053-23300/AEAS. **A duplicate copy of this sheet is enclosed.**

Please address all correspondence in connection with this application to:

Eric A. Stephenson
Conley, Rose & Tayon, P.C.
P.O. Box 398
Austin, TX 78767-0398
Ph: (512) 476-1400

Respectfully submitted,



Eric A. Stephenson
Reg. No. 38,321
Attorney for Applicants

Date:

7/30/99

PATENT
5053-23300

"EXPRESS MAIL" MAILING

LABEL NUMBER :

EL06365872US

DATE OF DEPOSIT

I HEREBY CERTIFY THAT THIS

PAPER OR FEE IS BEING

DEPOSITED WITH THE UNITED

STATES POSTAL SERVICE

"EXPRESS MAIL POST OFFICE TO

ADDRESSEE" SERVICE UNDER 37

C.F.R. § 1.10 ON THE DATE

INDICATED ABOVE AND IS

ADDRESSED TO THE

COMMISSIONER OF PATENTS

AND TRADEMARKS,

WASHINGTON, D.C. 20231

Debra J. Tix

“Event-Triggered Transaction Processing for Electronic Data Interchange”

By:

Joseph Frutuoso

David Nadeau

Robert C. Larew

660E20 99059E50

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to computer software, and more particularly to electronic transaction processing between distributed computer systems.

2. Description of the Related Art

Electronic data interchange (EDI) generally refers to the process of transmitting and/or receiving data in a predetermined digital format from one computer system to another computer system. EDI can be a fast, inexpensive, and relatively safe method of processing business-related transactions. For example, EDI can be used to send purchase orders, invoices, shipping notices, and other frequently used electronic business documents. EDI can greatly reduce errors in transaction processing by eliminating the need for manual entry of data in hardcopy business documents. EDI is therefore an advantageous method of processing business transactions, and businesses in particular fields such as insurance often expect to communicate with each other electronically.

In order for their computers to communicate effectively with one another, EDI participants must follow standards for the format and transmission of electronic data. Several standards maintenance organizations have set forth EDI standards. Examples of such standards and the associated maintenance organizations include ANSI X12, which was developed by the American National Standards Institute; the Uniform Communications Standards (UCS); the TDCC standard developed by the Transportation Data Coordinating Committee; and UN-EDIFACT (EDI for Administration, Commerce, and Transport), an international standard based on ANSI X12 and Trade Data Interchange standards commonly used in Europe.

The National Securities Clearing Corporation (NSCC) is another organization that promulgates standards for EDI. The NSCC maintains the Annuity Processing Service

(APS), an insurance industry clearinghouse which provides a flexible system for the transmission of information to and from trading partners such as insurance carriers, broker/dealers, agents, banks, and other organizations. The Annuity Processing Service may also be referred to as the Insurance Processing Service (IPS). Trading partners who are members of the NSCC may use the APS clearinghouse, for example, to transmit annuity information and clear associated monies so that transactions can be finalized quickly. As used herein, "trading partners" refer to business entities who transact business with one another through electronic data interchange. A "sending trading partner" is a trading partner who electronically sends data related to a transaction. Often, the sending trading partner is the trading partner who initiates the transaction. A "receiving trading partner" is a trading partner who electronically receives data related to a transaction. For some transactions, a sending trading partner and a receiving trading partner may be the same entity.

To enable the electronic processing of transactions through the APS, the NSCC provides standards for data formats for a variety of common transactions. As used herein, a "transaction" is any business event that is processed electronically. Transactions, including those supported by the APS, include business events such as deposits, applications, commission payments or settlements, positions and valuations (full value and refreshed value), annuity asset pricing, financial activity reporting, and other suitable events. Positions and valuations full refresh (PVF) is a transaction involving the financial and non-financial information about an annuity contract at a particular point in time. The PVF includes contract data such as valuations, replacements, producer information, owner information, and payor information. Positions and valuations focused refresh (PFF) is also a transaction involving the financial and non-financial information about an annuity contract at a particular point in time. The PFF record is a shorter version of the PVF record and includes the data and value for each contract. Annuity asset pricing (AAP) is a transaction involving the pricing or unit value of the underlying variable investment funds supporting a contract.

09365066-073099

The APS clearinghouse enables trading partners to transmit data to a variety of other trading partners without establishing proprietary networks. However, the adherence of trading partners to industry standard data formats is not enough to automate all transactions. Existing approaches often require the intervention of a skilled computer programmer for certain tasks and transactions. For example, in order to process a transaction such as a commission payment request received electronically from a broker/dealer, an insurance carrier often must write, test, and install a custom program in a programming language such as COBOL to tell its payment system to obtain the necessary broker-related information from another source, such as a particular database. This approach towards EDI tends to be wasteful, time-consuming, and expensive.

For at least the foregoing reasons, there is a need for an improved system and method for processing transactions in a more efficient manner.

SUMMARY OF THE INVENTION

The present invention provides various embodiments of a method and system for automating data exchange processing. In one embodiment, a trading relationship between trading partners is stored in a trading relationship database. Trading partners may include agents, broker/dealers, banks, and other suitable entities. In the insurance field, for example, trading partners may include insurance companies, insurance agents, and insurance broker/dealers. At least one trading partner is a sending trading partner and at least one trading partner is a receiving trading partner with respect to a transaction between the at least one sending trading partner and the at least one receiving trading partner. Maps and business rules may also be defined and stored with regard to particular types of transactions and particular trading partners. Maps define sources of additional data used to process a transaction, such as fields within an administration system. Business rules determine conditions and criteria to be met by additional data from an administration system specified in a map. In one embodiment, the maps are created with a drag-and-drop mapping of incoming transactions to outgoing transactions in a graphical user interface. The rules may also be defined by a user in a user interface.

09/05/2019 09:05:59

An incoming transaction may be received from the at least one sending partner as identified in a trading relationship. In one embodiment, the incoming transaction may be received through the Annuity Processing Service of the NSCC. The receiving of incoming transactions may take place according to a schedule.

In response to receiving the incoming transaction from the at least one sending trading partner, additional information may be read from an administration system in order to complete the processing of the transaction. One or more maps may determine the source of the additional data within the administration system. One or more business rules may be executed against records in the administration system to specify the additional information from the administration system. In various embodiments, the business rules may include combinations of keywords and/or logical operators. Keywords may include terms such as sending trading partner identifiers, receiving trading partner identifiers, administration system identifiers, transaction identifiers, and transaction statuses. Logical operators may include "AND", "OR", "NOT", "EQUALS", "NOT EQUAL TO", "GREATER THAN", and "LESS THAN". The business rule may include a string of at least one keyword and at least one logical operator which is constructed by a user of the computer system. The business rule may then be entered into the computer system by a user via a user interface and subsequently stored in a database. In one embodiment, the business rule, once constructed, is entered much like one would enter a search string into a search engine. The business rules may therefore function as search criteria for additional information to be obtained to complete a transaction. By allowing a user to construct and enter different business rules which are subsequently used to generate different outgoing transactions, the system and method for event-triggered transaction processing is adjustable to process an indefinite number of types of transactions without the intervention of a programmer writing specialized programs.

In one embodiment, the additional data may be obtained from more than one administration system. The administration system may be an in-house administration system such as an annuity administration system. In one embodiment, the administration

system is the VANTAGE-ONE® annuity administration system available from Computer Sciences Corporation. Additional information may be obtained according to a schedule. The schedule may include a particular time and date or a frequency. The incoming transaction may therefore include a triggering event which prompts the obtaining of additional data.

An outgoing transaction is generated in response to obtaining additional information from an administration system. The outgoing transaction may include the additional data obtained from the administration system. The outgoing transaction is sent to the at least one receiving trading partner. The sending of outgoing transactions may take place according to a schedule which may specify a date and time or a frequency at which outgoing transactions are to be sent. The outgoing transaction may be queued for sending along with other outgoing transactions. The outgoing transaction may be reformatted into an industry standard data format by an adapter. In one embodiment, the outgoing transaction may be sent through the Annuity Processing Service of the NSCC, and the outgoing transaction may be reformatted into an NSCC-standard data format.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a network diagram of an illustrative distributed computing environment which is suitable for implementing various embodiments;

Figure 2 is an illustration of a typical computer system which is suitable for implementing various embodiments;

Figure 3 is a block diagram illustrating an overview of the event-triggered transaction processing according to one embodiment;

Figure 4 is a block diagram illustrating an overview of the event supply process according to one embodiment;

Figure 5 is a block diagram illustrating an overview of the schedule supply process according to one embodiment;

Figure 6 is a block diagram illustrating an overview of the event process and the event polling process according to one embodiment;

Figure 7 is a block diagram illustrating an overview of the controller process and the rules process according to one embodiment;

Figure 8 is a block diagram illustrating an overview of the mapping process and the output process according to one embodiment;

Figure 9 is a flowchart illustrating an overview of event-triggered transaction processing according to one embodiment;

Figure 10 is a flowchart illustrating the storing of a trading relationship according to one embodiment;

Figure 11 is a flowchart illustrating the storing of a map according to one embodiment.

While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

DETAILED DESCRIPTION OF SEVERAL EMBODIMENTS

Figure 1: A Distributed Computing Environment

Figure 1 illustrates a distributed or enterprise computing environment according to one embodiment. A distributed computer system or enterprise 100 includes a plurality of computer systems which are interconnected through one or more networks. Although one particular embodiment is shown in Figure 1, the distributed computer system 100 may include a variety of heterogeneous computer systems and networks which are interconnected in a variety of ways and which run a variety of software applications and/or operating system software.

One or more local area networks (LANs) 104 may be included in the enterprise 100. A LAN 104 is a network that spans a relatively small area. Typically, a LAN 104 is confined to a single building or group of buildings. Each node (i.e., individual computer system or device) on a LAN 104 preferably has its own CPU with which it executes programs, and each node is also able to access data and devices anywhere on the LAN 104. The LAN 104 thus allows many users to share devices (e.g., printers) as well as data stored on file servers. The LAN 104 may be characterized by any of a variety of types of topology (i.e., the geometric arrangement of devices on the network), of protocols (i.e., the rules and encoding specifications for sending data, and whether the network uses a peer-to-peer or client/server architecture), and of media (e.g., twisted-pair wire, coaxial cables, fiber optic cables, radio waves). As illustrated in Figure 1, the distributed computer system 100 may include one LAN 104. However, in alternate configurations the distributed computer system 100 may include a plurality of LANs 104 which are coupled to one another through a wide area network (WAN) 102. A WAN 102 is a network that spans a relatively large geographical area.

Each LAN 104 includes a plurality of interconnected computer systems and optionally one or more other devices: for example, one or more workstations 110a, one or more personal computers 112a, one or more laptop or notebook computer systems 114, one or more server computer systems 116, and one or more network printers 118. As

illustrated in Figure 1, an example LAN 104 may include one of each of computer systems 110a, 112a, 114, and 116, and one printer 118. The LAN 104 may be coupled to other computer systems and/or other devices and/or other LANs 104 through a WAN 102.

One or more mainframe computer systems 120 may be coupled to the distributed computer system 100. As shown in Figure 1, the mainframe 120 may be coupled to the distributed computer system 100 through the WAN 102, but alternatively one or more mainframes 120 may be coupled to the distributed computer system 100 through one or more LANs 104. As shown, the mainframe 120 may be coupled to a storage device or file server 124 and mainframe terminals 122a, 122b, and 122c. The mainframe terminals 122a, 122b, and 122c may access data stored in the storage device or file server 124 coupled to or included in the mainframe computer system 120.

The distributed computer system 100 may also include one or more computer systems which are connected to the distributed computer system 100 through the WAN 102: as illustrated, a workstation 110b and a personal computer 112b. In other words, the enterprise 100 may optionally include one or more computer systems which are not coupled to the distributed computer system 100 through a LAN 104. For example, the distributed computer system 100 may include computer systems which are geographically remote and connected to the distributed computer system 100 through the Internet.

In one embodiment, the enterprise or distributed computer system 100 includes computer systems for a plurality of trading partners, an industry clearinghouse system, and a computer system for transaction processing. The computer systems for the trading partners, for the industry clearinghouse, and for transaction processing may be computer systems as shown in Figure 1 and may be coupled to one another through a WAN 102. The trading partners and computer system for transaction processing are configured to exchange transaction data electronically with one another through the industry clearinghouse. The industry clearinghouse may require transaction data to be exchanged

in a particular data format. For example, in one embodiment the industry clearinghouse is the Annuity Processing System (APS) or Insurance Processing System (IPS) of the National Securities Clearing Corporation (NSCC). In one embodiment, the computer system for transaction processing includes industry adapters to convert or translate incoming transaction data from NSCC data formats or other standard data formats and outgoing transaction data to NSCC data formats or other standard data formats. As used herein, an "industry adapter" includes a computer program, utility, driver, or interface which translates or converts data to or from a standardized data format.

Figure 2: A Typical Computer System

Figure 2 illustrates a typical computer system 150 which is suitable for implementing various embodiments. Each computer system 150 typically includes components such as a CPU 152, with an associated memory medium such as floppy disks 160. The memory medium may store program instructions for computer programs, wherein the program instructions are executable by the CPU 152. The computer system 150 may further include a display device such as a monitor 154, an alphanumeric input device such as a keyboard 156, and a directional input device such as a mouse 158. The computer system 150 is operable to execute the computer programs to implement event-triggered transaction processing as described herein.

The computer system 150 preferably includes a memory medium on which computer programs according to various embodiments may be stored. The term "memory medium" is intended to include an installation medium, e.g., a CD-ROM, or floppy disks 160, a computer system memory such as DRAM, SRAM, EDO RAM, Rambus RAM, etc., or a non-volatile memory such as a magnetic media, e.g., a hard drive, or optical storage. The memory medium may include other types of memory as well, or combinations thereof. In addition, the memory medium may be located in a first computer in which the programs are executed, or may be located in a second different computer which connects to the first computer over a network. In the latter instance, the second computer provides the program instructions to the first computer for execution. Also, the computer system 150 may take various forms, including a personal computer system, mainframe computer system,

workstation, network appliance, Internet appliance, personal digital assistant (PDA), television system or other device. In general, the term "computer system" can be broadly defined to encompass any device having a processor which executes instructions from a memory medium.

The memory medium preferably stores a software program or programs for event-triggered transaction processing as described herein. The software program(s) may be implemented in any of various ways, including procedure-based techniques, component-based techniques, and/or object-oriented techniques, among others. For example, the software program may be implemented using ActiveX controls, C++ objects, JavaBeans, Microsoft Foundation Classes (MFC), or other technologies or methodologies, as desired. A CPU, such as the host CPU 152, executing code and data from the memory medium includes a means for creating and executing the software program or programs according to the methods and/or block diagrams described below.

The computer system for transaction processing may be a typical computer system 150 as illustrated in Figure 2. As will be discussed in greater detail below, the computer system for transaction processing is operable to receive incoming transaction data from another computer system, process the transaction data, and generate and send outgoing transaction data to another computer system. In one embodiment, transaction data is received from trading partners and sent to trading partners through an industry clearinghouse system. In one embodiment, the NSCC's Annuity Processing Service (APS) clearinghouse is used as an industry clearinghouse for exchange of the transaction data. As used herein, transactions may include business events such as deposits, applications, commission payments or settlements, positions and valuations (full value and refreshed value), annuity asset pricing, insurance pricing, securities reporting, financial activity reporting, and other suitable events supported by the APS clearinghouse. Transactions may generally include insurance-related transactions. As used herein, an "insurance-related transaction" is a transaction involving insurance policies, contracts, funds, valuations, companies, agencies, agents, or broker/dealers. The

computer system for transaction processing may be maintained by an organization such as an insurance carrier.

In one embodiment, the computer system for transaction processing is coupled to one or more administration systems. Administration systems may include, for example, database management systems or annuity administration systems such as VANTAGE-ONE from Computer Sciences Corporation. The administration systems may be in-house systems, external systems, or a combination of in-house and external systems.

As used herein, an "administration system" includes any computer-based system that provides data needed for a transaction. If the transaction is a positions and valuations full refresh (PVF) transaction, for example, then data needed to complete the PVF transaction may include contract data or policy information such as valuations, replacements, producer, owner, and payor information. The administration system may contain the policy information necessary to complete this PVF transaction. As another example, if the transaction is an annuity asset pricing (AAP) transaction, then the computer system for transaction processing is expected to tell the trading partner the pricing or unit values of the underlying investment funds supporting a contract. This pricing is required to calculate the dollar value of an annuity contract and financial transactions. Typically, the records are generated each business day and contain information for each investment fund used by a company. An annuity administration system such as VANTAGE-ONE may include the data required to create and support AAP transactions. For example, the administration system may include a fund information table which provides details about the investment fund. The administration system may also include a unit value history file which contains the unit values for each business day that prices are available.

Figure 3: Overview of Event-Triggered Transaction Processing

Figure 3 is a block diagram showing an overview of the system and method for event-triggered transaction processing. As shown in figure 3, according to one embodiment, an initial configuration process may include the initialization 300 of a set of tables with appropriate data. These tables may include a system registry table, a file

registry table, an event registry, a trading partner table, a trading relationship table, and a scheduled events table. In other embodiments, these and other tables may be initialized at other points in time, at or before the time that the data in the tables is needed by the system and method for event-triggered transaction processing. These and other tables may be implemented and stored in a database. As used herein, a "database" is a computer-based system for storing and retrieving data. The contents of these tables are discussed below.

In one embodiment, the relationship of several processes and tables is as follows. When an incoming transaction is received from a sending trading partner, a corresponding event is generated. As used herein, an "event" is a computer record corresponding to a business transaction or an action taken in accordance with a business transaction. The event supply process 302 passes events to the event process 306. The schedule supply process 304 passes schedules for events to the event process 306. The event process 306 populates the events table 308 with the events and schedules. The event polling process 310 reads events and schedules from the events table 308 and queues events to the queued events table 312. The controller process 314 reads data from the queued events table 312 and initiates the rules process 320, the mapping process 316, and the output process 318. The event polling process 310 provides the ability to determine which events are selected and controls when these events are queued for mapping. The controller process 314 directs the processing of events by applying the appropriate rules and mapping instructions. A data mapper may define and build rules to be used by the mapping process 316. The mapping process 316 carries out or executes the rules defined and built in the data mapper. By applying user-defined business rules in this way, additional data may be obtained from administration systems to complete transactions. The rules process 320 interprets the rules and determines if processing of events or maps may continue. The output process 318 provides a central storage area for all records processed, including outgoing transactions built in the preceding processes. The outgoing transaction may be sent through an industry adapter 322 to the receiving trading partner, directly or indirectly through an industry clearinghouse. These processes and tables are discussed in greater detail below.

Figure 4: Event Supply Process

Figure 4 is a block diagram illustrating the event supply process according to one embodiment. The event supply process 302 accesses the file registry table 410 using parameter input 400 to verify that a requested file has been registered. The file registry table 410 contains descriptions of files needed for processing. Files may store events, transactions, and other data related to transaction processing. The file registry table may contain, for each file, a file name, a system identification number, a file type (e.g., a database type such as VSAM, sequential, or DB2), and an application programming interface (API) which names the module that reads the file to create events.

Parameter input 400 may include information relating to a requested file. A match is the condition that the requested file exists on the file registry table 410. If a match is not found, error processing may begin. Error processing is indicated in the Figures by an [E] in a lower corner of a block diagram box to indicate that, in one embodiment, the system for event-triggered transaction processing has centralized error processing for records and events that fail. When an error is encountered, the system for event-triggered transaction processing analyzes it to determine the severity of the error. The error is then recorded in the error log, and the system for event-triggered transaction processing responds according to the severity of the error. If the error is a warning, processing can continue, and the warning message indicates which record had the problem. For a more severe error, processing stops, and an error message provides an explanation. If the error is fatal, the system for event-triggered transaction processing abends (e.g., shuts down).

If a match is found on the file registry table 410, the file type and application programming interface (API) for the file are retrieved from the file registry table 410. Next, the event supply process 302 opens and reads records from the file. For each record, the API 404 is called. In one embodiment, the event supply process 302 passes the data record along with an instruction to build the event to the API 404. The API 404 discards any unwanted records and returns an event string 408 to be passed to the event process 306.

Figure 5: Schedule Supply Process

Figure 5 is a block diagram illustrating the schedule supply process according to one embodiment. In one embodiment, the schedule supply process 304 accesses the scheduled events table 502 using parameters 500 to determine the events to be processed. The scheduled events table 502 identifies the business events that will be created by the schedule supply process. The scheduled events table 502 may contain information relating to schedules for events. Schedules for events may include a sending trading partner identifier such as an identification number to identify a sending trading partner, an administration system identifier such as an identification number, an output event identifier such as a name, an input event identifier such as a name, a date and/or time the event is scheduled to be processed, an event frequency (e.g., daily, weekly, or monthly), an event frequency increment, a current/all indicator, a status (e.g., active or suspended), and an application programming interface which names the module that reads the file to create events. The event frequency increment specifies a frequency at which the event should be run in event frequency units. For example, to schedule an event every other week, the event frequency should be set to "weekly," and the event frequency increment should be set to "2." The current/all indicator identifies whether all schedules for a given event should be run or only the current schedule should be run.

The trading relationship table 504 is accessed to determine if active trading partner relationships exist for the scheduled events records that have been found. The trading relationship table 504 stores data related to trading relationships. As used herein, a "trading relationship" includes a relationship between one or more sending trading partners and one or more receiving trading partners. A trading relationship may also include the status of the relationship (e.g., active, inactive, or suspended), information related to the event being sent from the sending trading partner to the receiving trading partner, a trading partner identifier such as an identification number for a sending trading partner, a trading partner identifier such as an identification number for a receiving trading partner, an output event identifier such as an output event name, a destination for the event, an event frequency (e.g., daily, weekly, or monthly), a trading relationship status (e.g., active, inactive, or suspended), and an indicator (e.g., test or production).

For each record (i.e., event) with an active trading partner relationship in the trading relationship table 504, an application programming interface (API) 506 to an appropriate administration system is retrieved from the scheduled events table 502. When API 506 is called, it returns an event string 508 to be passed to the event process. Based on the type of scheduled event or transaction, the API 506 may be used in different ways. For an annuity asset pricing (AAP) transaction, for example, the API 506 may be used to access a fund information file for processing each record (i.e., fund) that is active. The API 506 may also be used to access the unit value history field to obtain fund values for AAP transactions. For a positions and valuations focused refreshed (PFF) transaction or a positions and valuations full refresh (PVF) transaction, the API 506 may be used to access a policy hierarchy record using a receiving trading partner identifier from the trading relationship table 504 as an agent identifier. For PFF and PVF transactions, the API 506 may also be used to access a policy commission record for each policy fund. In one embodiment, the API 506 is used to read, extract, or otherwise obtain additional information from an administration system according to a map and/or business rules and/or search criteria, as discussed in greater detail below with reference to the mapping process 316.

The records retrieved from the scheduled events table 502 are updated based on frequency and increment to the next scheduled date. If multiple records are found on the scheduled events table 502, each one is examined separately until all records are processed. The schedule supply process 304 may also include both error processing and activity log/tracking (indicated in the Figures by an [A] in a lower corner of a block diagram box). In one embodiment, the scheduled events table 502 is scanned daily, and an error message is generated if no records (i.e., events) are found on a given day. In activity log/tracking, the system for event-triggered transaction processing logs the start of a process, the in-progress stage of the process, and the completion of the process. For balancing and verification purposes, the following totals may be provided by activity log/tracking: number of records input, number of records discarded, number of errors, and number of events created.

Figure 6: Event and Event Polling Processes

Figure 6 is a block diagram illustrating the event process 306 and the event polling process 310 according to one embodiment. Both of these processes may have error processing and activity log/tracking, as described with reference to Figures 4 and 5, respectively. In one embodiment, the event process 306 validates the transmitting administration system by comparing the system identification number on the incoming data 408 from the event supply process and the incoming data from the schedule supply process 508 to the system registry table 600. The system registry table 600 contains information regarding the transmitting administration system of the sending trading partner. The system registry table 600 may contain identifiers such as an identification number and name for each transmitting administration system. If the administration system is one that will be accessed to obtain additional data to complete a transaction, the system registry table 600 may also contain an application program interface which identifies the module that accesses the administration system, as discussed with reference to Figure 5.

In one embodiment, the event process 306 creates events and updates the events table 308. An event corresponds to one or more actions required to process a transaction. The events table 308 contains created events. The events table 308 may contain a record for each event, wherein the record may include a sending trading partner identifier such as an identification number, an administration system identifier such as an identification number, a receiving trading partner identifier such as an identification number, an input event identifier such as a name, an output event identifier such as a name, and an event timestamp. The events table 308 may also contain data specific to a type of transaction. For example, a record for an PFF or PVF transaction in the events table 308 may include a company code, a policy number, a plan code, and an agent ID. A record for an AAP transaction may include a company code, a fund number, a fund type, and a fund value.

The event process 306 may validate the events by accessing the event registry table 602. The event registry table 602 contains information about the sending trading

partner and administration system identifiers for a particular event. The event process 306 may also validate that the trading relationship table 504 contains an active record for the pair of partners in question.

In one embodiment, the event polling process 310 accesses the poller table 604 to determine which events require further processing, selects the appropriate events from the events table 308, and enters these events on the queued events table 312 with a status of "queued." The poller table 604 establishes the selection criteria to identify the events that should be queued for output. The poller table 604 may contain a poller identifier such as an identification number, a queue identifier such as an identification number, a sending trading partner identifier such as an identification number, a receiving trading partner identifier such as an identification number, an administration system identifier such as an identification number, an output event identifier such as a name, and an event status. The queued events table 312 may contain a queue identifier such as an identification number, a sending trading partner identifier such as an identification number, an administration system identifier such as an identification number, a receiving trading partner identifier such as an identification number, an input event identifier such as a name, an output event identifier such as a name, and an event status.

Figure 7: Controller and Rules Processes

Figure 7 is a block diagram illustrating the controller process 314 and the rules process 320 according to one embodiment. Both of these processes may have error processing and activity log/tracking, as described with reference to Figures 4 and 5, respectively. In one embodiment, the controller process 314 verifies that a mapset and a ruleset for a particular event exist on the process table 702. The process table 702 contains the sending trading partner and receiving trading partner identifiers such as identification numbers, input and output event identifiers such as names, application program interfaces, and the mapset and ruleset to complete mapping and rules processing. A mapset and ruleset may be used to generate an outgoing transaction. A mapset is a collection of one or more maps. A map is a relationship between one or more source administration systems and one or more destination administration systems. A ruleset is a collection of one or more

business rules. As used herein, a “business rule” is one or more terms requiring that specified conditions or criteria be met in retrieving additional data from an administration system. Maps and business rules are discussed in greater detail below.

The controller process 314 may call the rules process 320 to apply any business rules that have been assigned. The rules process 320 obtains the rules from the ruleset table 704 and determines if each rule is true or false with respect to each of any number of records of additional data retrieved from an administration system. The ruleset table 704 contains one or more rules associated with a ruleset. The details of each rule may be found in the rules table 706. An individual rule may contain one or more keywords designed to retrieve the appropriate additional data to complete a transaction. The details of each keyword may be found in the keyword table 708. If a rule is found to be true by the rules process 320, processing can continue. If a rule is found to be false by the rules process 320, the appropriate record is bypassed.

The controller process 314 calls the mapping process 316 to apply the mapping instructions from the mapset. The mapping process 316 is discussed in greater detail with respect to Figure 8. The controller process 314 may also determine the output administration system to which the event will be sent. The controller process calls the output process 318 to create the output files, wherein the output files include outgoing transactions.

Figure 8: Mapping and Output Processes

Figure 8 is a block diagram illustrating the mapping process 316 and the output process 318 according to one embodiment. Both of these processes may have error processing and activity log/tracking, as described with reference to Figures 4 and 5, respectively. A data mapper 800 is a software application which permits a user to define one or more maps or mapsets for use by the mapping process 316. In one embodiment, the data mapper 800 displays source and destination record fields and permits a user to establish a connection from one or more fields in the source record to a field in the destination record. In this way, the user may define a map which determines the source

of additional data to be obtained from an administration system to generate an outgoing transaction. In order to display the source and destination record fields, a map name, a source administration system, a file name for the source administration system, a destination administration system, and a file name for the destination administration system must be chosen in one embodiment.

The data mapper 800 may include a graphical user interface (GUI). With a GUI according to one embodiment, source fields may be selected to create a source selection. The source selection may then be dragged and dropped onto a destination field. Additionally, functions may be applied to transform or convert either the source or the destination fields. If functions are not applied, and more than one source field is mapped to a destination field, then by default the value of the destination field is the sum of the values of the mapped source fields.

Business rules may also be defined and built in the data mapper application 800. In one embodiments, the business rules may include combinations of keywords and/or logical operators. Keywords may include terms such as sending trading partner identifiers, receiving trading partner identifiers, administration system identifiers, transaction identifiers, and transaction statuses. Logical operators may include "AND", "OR", "NOT", "EQUALS", "NOT EQUAL TO", "GREATER THAN", and "LESS THAN". The business rule may include a string of at least one keyword and at least one logical operator which is constructed by a user of the computer system. The business rule may then be entered into the computer system by a user via a user interface and subsequently stored in a database. In one embodiment, the business rule, once constructed, is entered much like one would enter a search string into a search engine. The business rules may therefore function as search criteria for additional information to be obtained to complete a transaction. The business rules may use or reuse existing rules, keywords, and/or functions to build new rules or functions for obtaining the additional data and completing transactions. By allowing a user to construct and enter different business rules which are subsequently used to generate different outgoing transactions, the system and method for event-triggered transaction processing is adjustable to process

an indefinite number of types of transactions without the intervention of a programmer writing specialized programs.

The data mapper 800 may access the following mapping-related tables: function table 802, function parameters table 804, map name table 806, map field table 808, map ID table 810, map function on input table 812, map function on output table 814, rules table 706, and keyword table 708. The function table 802 may contain functions that can be applied to a field. The function parameters table 804 may contain a list of parameters associated with the functions. The map name table 806 may contain a list of available maps. The map field table 808 may define all the fields in a map. The map ID table 810 may define how the fields are mapped from the input (source) to the output (destination). The map function on input table 812 may define a function applied to an input field. The map function on output table 814 may define a function applied to an output field. The keyword table 708 may contain the keywords used in the mapping process.

After maps and rules have been defined and built using the data mapper 800, the mapping process 316 may execute those maps and rules. The mapping process 316 applies one or more business rules to retrieve additional data from an administration system specified in a map. To determine how to generate the outgoing transaction, the mapping process 316 may access the following mapping tables: mapset table 816, map ID table 810, map function on input table 812, and map function on output table 814. The mapset table 816 may contain one or more mapsets. In one embodiment, the input data is obtained from the source administration system; functions are applied, if appropriate; and the data is mapped to the output table 820 via the output process 318. The output table 820 may contain the outgoing record (i.e., the outgoing transaction) as well as information on the sending and receiving trading partners, the type of record, and date and time the record was processed.

The output process 318 may provide a central storage area for all processed transactions. In one embodiment, records are stored in a common format and remain in this storage area until being translated by an appropriate industry adapter 322. The

output process 318 receives output data from the mapping process 316. For each output record received, the output process 318 may then call the trading partner table 818 to obtain identifying information. The trading partner table 818 may contain identifiers such as the names and identification numbers of the sending and receiving trading partners. The output process 318 then builds an information header for each record and writes the record to the output table 820 with a status of "new." After each outgoing transaction is processed by an industry adapter 322, the output process 318 changes the status of the corresponding record in the output table 820 to "complete."

The industry adapter 322 may determine which records should be in a test file and which records should be in a production file. In one embodiment, the industry adapter 322 may attach headers and/or trailers required by the NSCC to outgoing transactions. In one embodiment, AAP records are sent to the NSCC, which passes the appropriate information to each of the company's trading partners listed on a trading partner profile maintained at NSCC.

Figure 9: An Overview of Event-Triggered Transaction Processing

Figure 9 is a flowchart of an embodiment of a system and method for event-triggered transaction processing. In step 900, a trading relationship between trading partners is stored in a trading relationship database. Trading partners may include agents, banks, insurance companies, broker/dealers, and other suitable entities. In the insurance field, for example, trading partners may include insurance companies, insurance agents, and insurance broker/dealers. At least one trading partner is a sending trading partner and at least one trading partner is a receiving trading partner with respect to a transaction between the at least one sending trading partner and the at least one receiving trading partner. In step 900, maps and business rules may also be defined and stored with regard to particular types of transactions and particular trading partners. In one embodiment, the maps and rules are created with a drag-and-drop mapping of incoming transactions to outgoing transactions in a graphical user interface, as discussed with reference to the data mapper application 800 in Figure 8.

In step 901, an incoming transaction is received from the at least one sending partner as identified in a trading relationship. In one embodiment, the incoming transaction may be received through an industry clearinghouse such as the Annuity Processing Service of the NSCC. The receiving of incoming transactions may take place at a predetermined time according to a schedule. The schedule may include a particular time and date or a frequency.

In step 902, in response to receiving the incoming transaction from the at least one sending trading partner, additional information is read, obtained, or otherwise extracted from an administration system in order to complete the processing of the transaction. One or more maps may determine the source of the additional data to be obtained. One or more business rules may be executed against records in the administration system to specify particular output from the administration system. In one embodiment, the additional data may be obtained from more than one administration system. The administration system may be an in-house administration system such as an annuity administration system. In one embodiment, the administration system is the VANTAGE-ONE annuity administration system available from Computer Sciences Corporation. Additional information may be obtained at a predetermined time according to a schedule. The schedule may include a particular time and date or a frequency. The incoming transaction may therefore include a triggering event which prompts the obtaining of additional data.

In step 903, an outgoing transaction is generated in response to obtaining additional information from an administration system. The outgoing transaction may include the additional data obtained from the administration system. In step 904, the outgoing transaction is sent to the at least one receiving trading partner. The sending of outgoing transactions may take place at a predetermined time according to a schedule which may specify a date and time or a frequency at which outgoing transactions are to be sent. The outgoing transaction may be queued along with other outgoing transactions for sending at a later time. The outgoing transaction may be reformatted into an industry standard data format by an adapter. In one embodiment, the outgoing transaction may be

sent through the Annuity Processing Service of the NSCC, and the outgoing transaction may be reformatted into an NSCC-standard data format.

Figure 10: Storing a Trading Relationship

Figure 10 shows steps that may further illustrate storing a trading relationship between trading partners, as shown by step 900 in Figure 9, according to one embodiment. The steps shown in Figure 10 may be performed in any order, in various embodiments, including simultaneously. In step 920, at least one trading partner is defined as a sender and at least one trading partner is defined as a receiver. In step 921, an output event or transaction is created which is associated with an incoming event from the sending trading partner. In step 922, a destination describing the location of the receiving trading partner is established. In step 923, an event frequency describing the frequency at which a scheduled event is to be run is established. In step 924, a status indicator, describing the condition of the relationship between the sending trading partner and the sending trading partner as it relates to the destination, is established. In step 925, a test/production indicator, describing the type of run of the transaction processing, is established. In step 926, an output format is established for an event.

Figure 11: Storing a Map

Figure 11 shows steps that may further illustrate storing a map, as shown by step 900 in Figure 9, according to one embodiment. In various embodiments, the steps shown in Figure 11 may be performed in a different order than the order shown in Figure 11. For example, steps 930 through 934 may be performed in any order, including simultaneously. In step 930, a map name is specified. In step 931, a source administration system is selected. In step 932, an associated source file name is selected. In step 933, a destination administration system is selected. In step 934, an associated destination file name is selected. In step 935, a source selection, which is a selection of one or more source fields, is created through user interaction with a graphical user interface (GUI). In step 936, the source selection is dragged to a destination field through the GUI. In step 937, the source selection is dropped on the destination field through the GUI.

The value of the destination field may be determined by the existence of a function. In step 938, it is determined whether a function should be applied and/or what type of function should be applied. In step 939, no function is applied and the value of the destination field is determined to be the sum of the values of the selected source fields. In step 940, a source-side function is applied to one or more of the source fields in the source selection, and the value of the destination field is determined to be the resulting value of the sum of the values of the selected source fields after application of the source-side function. In step 941, a destination-side function is applied to one or more of the destination fields, and the value of the destination field is determined to be the resulting value of first summing the values of the selected source fields and then applying the destination side function.

Various embodiments further include receiving or storing instructions and/or data implemented in accordance with the foregoing description upon a carrier medium. Suitable carrier media include storage media or memory media such as magnetic or optical media, e.g., disk or CD-ROM, as well as signals such as electrical, electromagnetic, or digital signals, conveyed via a communication medium such as networks 102 and/or 104 and/or a wireless link.

Although the system and method of the present invention have been described in connection with several embodiments, the invention is not intended to be limited to the specific forms set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents as can be reasonably included within the spirit and scope of the invention as defined by the appended claims.

What is claimed is:

1. A method comprising:
a computer system receiving an incoming transaction from at least one sending trading partner;
the computer system reading additional information from an administration system in data communication with the computer system, wherein the additional information is read in response to the computer system receiving an incoming transaction from the at least one sending trading partner, and wherein the additional information is identified by at least one business rule;
the computer system generating an outgoing transaction in response to reading the additional information from the administration system;
the computer system sending the outgoing transaction to at least one receiving trading partner.
2. The method of claim 1,
wherein the at least one business rule comprises one or more keywords.
3. The method of claim 1,
wherein the at least one business rule comprises one or more logical operators.
4. The method of claim 1,
wherein the business rule comprises a string of at least one keyword and at least one operator, and wherein the business rule is entered into the computer system by a user via a user interface.
5. The method of claim 1,
wherein the outgoing transaction comprises the additional information read from the administration system.

6. The method of claim 1,
wherein the reading additional information from the administration system
in response to the computer system receiving the incoming
transaction from the at least one sending trading partner further
comprises:
extracting the additional information from the administration
system according to search criteria.
7. The method of claim 6,
wherein the search criteria comprise one or more keywords.
8. The method of claim 1, further comprising:
queuing the outgoing transaction in response to the computer system
generating the outgoing transaction.
9. The method of claim 1,
wherein the computer system receiving the incoming transaction from the
at least one sending trading partner further comprises the computer
system receiving the incoming transaction from the at least one
sending trading partner through an industry clearinghouse system.
10. The method of claim 1,
wherein the computer system sending the outgoing transaction to the at
least one receiving trading partner further comprises the computer
system sending the outgoing transaction to the at least one
receiving trading partner through an industry clearinghouse
system.
11. The method of claim 1, further comprising:

the computer system translating the outgoing transaction from a first data format to a second data format before sending the outgoing transaction to the at least one receiving trading partner, wherein the second data format is an industry standard data format.

12. The method of claim 1,
wherein the incoming transaction is an insurance-related transaction.

13. A system comprising:
a CPU;
a database coupled to the CPU;
an administration system coupled to the CPU;
a memory coupled to the CPU, wherein the memory stores one or more computer programs executable by the CPU;
wherein the computer programs are executable to:
store a trading relationship between trading partners of a transaction, wherein the trading relationship is stored in the database, wherein at least one trading partner is a sending trading partner and at least one trading partner is a receiving trading partner;
receive an incoming transaction from the at least one sending trading partner;
read additional information from the administration system in response to receiving the incoming transaction from the at least one sending trading partner, wherein the additional information is identified by at least one business rule;
generate an outgoing transaction in response to reading the additional information from the administration system;
send the outgoing transaction to the at least one receiving trading partner, wherein the at least one receiving trading partner is identified in the trading relationship.

14. The system of claim 13,
wherein the business rule comprises a string of at least one keyword and at least one operator, and wherein the business rule is entered into the computer system by a user via a user interface.
15. The system of claim 13,
wherein the at least one business rule is defined by a user through a user interface.
16. The system of claim 13,
wherein receiving the incoming transaction from the at least one trading partner further comprises receiving the incoming transaction from the at least one trading partner through an industry clearinghouse system.
17. The system of claim 13,
wherein the incoming transaction is an insurance-related transaction.
18. A carrier medium which stores program instructions, wherein the program instructions are executable by a computer system to implement the method of:
receiving an incoming transaction from at least one sending trading partner;
reading additional information from an administration system in response to receiving the incoming transaction from the at least one sending trading partner, wherein the additional information is identified by at least one business rule;
generating an outgoing transaction in response to reading the additional information from the administration system;
sending the outgoing transaction to at least one receiving trading partner.

19. The carrier medium of claim 18,
wherein the at least one business rule comprises one or more keywords.
20. The carrier medium of claim 18,
wherein the at least one business rule comprises one or more logical
operators.
21. The carrier medium of claim 18,
wherein the at least one business rule comprises a string of at least one
keyword and at least one operator, and wherein the business rule is
entered into the computer system by a user via a user interface.
22. The carrier medium of claim 18,
wherein the at least one business rule is stored in a database.
23. The carrier medium of claim 18,
wherein the administration system from which additional information is
read is specified by a map, wherein the map comprises a
relationship between the outgoing transaction and a source for the
additional information.
24. The carrier medium of claim 23,
wherein the map is specified by a user through a user interface.
25. The carrier medium of claim 23,
wherein the program instructions are further executable by the computer
system to implement generating the map, wherein generating the
map comprises:
selecting one or more source fields, wherein each source field
corresponds to the source for the additional information;

selecting a destination field, wherein each destination field corresponds to the outgoing transaction.

26. The carrier medium of claim 25,
wherein a value of the destination field is a sum of respective values of the one or more selected source fields.
27. The carrier medium of claim 25,
wherein the program instructions are further executable by the computer system to implement generating a value of the destination field as a function of the one or more source fields.
28. The carrier medium of claim 18,
wherein the outgoing transaction comprises the additional information read from the administration system.
29. The carrier medium of claim 18,
wherein the program instructions are further executable by the computer system to implement storing a schedule in memory, wherein the schedule relates to the incoming transaction, and wherein the schedule comprises a predetermined time for receiving the incoming transaction from the at least one sending trading partner.
30. The carrier medium of claim 18,
wherein the program instructions are further executable by the computer system to implement storing a schedule in memory, wherein the schedule relates to the incoming transaction, and wherein the schedule comprises a predetermined time for reading the additional information from the administration system.
31. The carrier medium of claim 18,

wherein the program instructions are further executable by the computer system to implement storing a schedule in memory, wherein the schedule relates to the outgoing transaction, and wherein the schedule comprises a predetermined time for sending the outgoing transaction to the at least one receiving trading partner.

32. The carrier medium of claim 18,
wherein reading additional information from the administration system in response to receiving the incoming transaction from the at least one sending trading partner further comprises extracting the additional information from the administration system according to search criteria.
33. The carrier medium of claim 32,
wherein the search criteria comprise one or more keywords.
34. The carrier medium of claim 18,
wherein the program instructions are further executable by the computer system to implement queuing the outgoing transaction in response to generating the outgoing transaction.
35. The carrier medium of claim 18,
wherein the incoming transaction is received from the at least one trading partner through an industry clearinghouse.
36. The carrier medium of claim 18,
wherein the outgoing transaction is sent to the at least one receiving trading partner through an industry clearinghouse.
37. The carrier medium of claim 18,

wherein the program instructions are further executable by the computer system to implement translating the outgoing transaction from a first data format to a second data format before sending the outgoing transaction to the at least one receiving trading partner, wherein the second data format is an industry standard data format.

38. The carrier medium of claim 18,
wherein the incoming transaction is an insurance-related transaction.
39. The carrier medium of claim 18,
wherein the outgoing transaction is an insurance-related transaction.
40. The carrier medium of claim 18,
wherein the outgoing transaction is an annuity asset pricing transaction.
41. The carrier medium of claim 18,
wherein the outgoing transaction is a positions and valuation focused refresh transaction.
42. The carrier medium of claim 18,
wherein the outgoing transaction is a positions and valuation full refresh transaction.
43. The carrier medium of claim 18,
wherein the outgoing transaction is an insurance pricing transaction.
44. The carrier medium of claim 18,
wherein the outgoing transaction is a commission settlement transaction.
45. The carrier medium of claim 18,
wherein the sending trading partner is the receiving trading partner.

46. The carrier medium of claim 18,
wherein the carrier medium is a memory medium.
47. A carrier medium which stores program instructions, wherein the program instructions are executable by a computer system to implement the method of:
storing a trading relationship between trading partners of a transaction,
wherein at least one trading partner is a sending trading partner and
at least one trading partner is a receiving trading partner;
receiving an incoming transaction from the at least one sending trading
partner;
reading additional information from an administration system in response
to receiving the incoming transaction from the at least one sending
trading partner, wherein the additional information obtained is
identified by at least one business rule;
generating an outgoing transaction in response to reading additional
information from the administration system;
sending the outgoing transaction to the at least one receiving trading
partner, wherein the at least one receiving trading partner is
identified in the trading relationship.
48. The carrier medium of claim 47,
wherein the at least one business rule comprises a string of at least one
keyword and at least one operator, and wherein the business rule is
entered into the computer system by a user via a user interface.
49. The carrier medium of claim 47,
wherein the at least one business rule is stored in a database.
50. The carrier medium of claim 47,

wherein the outgoing transaction is an insurance-related transaction.

51. The carrier medium of claim 47,
wherein the carrier medium is a memory medium.
52. The method of claim 1,
wherein the at least one business rule comprises a receiving trading
partner identifier.
53. The method of claim 1,
wherein the at least one business rule comprises an administration system
identifier.
54. The method of claim 1,
wherein the at least one business rule comprises a transaction identifier.
55. The method of claim 1,
wherein the at least one business rule comprises a transaction status.
56. The method of claim 1,
wherein the at least one business rule comprises a sending trading partner
identifier.
57. The method of claim 1,
wherein the business rule is entered into a database.
58. The carrier medium of claim 18,
wherein the at least one business rule comprises a receiving trading
partner identifier.
59. The carrier medium of claim 18,

wherein the at least one business rule comprises an administration system identifier.

60. The carrier medium of claim 18,
wherein the at least one business rule comprises a transaction identifier.
61. The carrier medium of claim 18,
wherein the at least one business rule comprises a transaction status.
62. The carrier medium of claim 18,
wherein the at least one business rule comprises a sending trading partner identifier.

09365066-073099
660E20-89059E60

ABSTRACT OF THE DISCLOSURE

A method and system for automating data exchange processing by using event-triggered transaction processing. Transaction data may be exchanged electronically with industry clearinghouses or in-house administration systems. A trading relationship between trading partners is stored in a trading relationship database. At least one trading partner is a sending trading partner and at least one trading partner is a receiving trading partner with respect to a transaction between the sender and receiver. Maps and rules may be created and stored with respect to transactions and/or trading partners. An incoming transaction is received through an industry clearinghouse from the at least one sending partner as identified in the trading relationship. In response to receiving the incoming transaction, additional information is read from an administration system specified by a map in order to complete the processing of the transaction. The incoming transaction is a triggering event which prompts the obtaining of additional data. The additional data may be obtained according to user-specified business rules. In response to obtaining additional information, an outgoing transaction which may include the additional data is generated and sent to the receiving trading partner through the industry clearinghouse.

09365066-073099
660E20" 99099E60

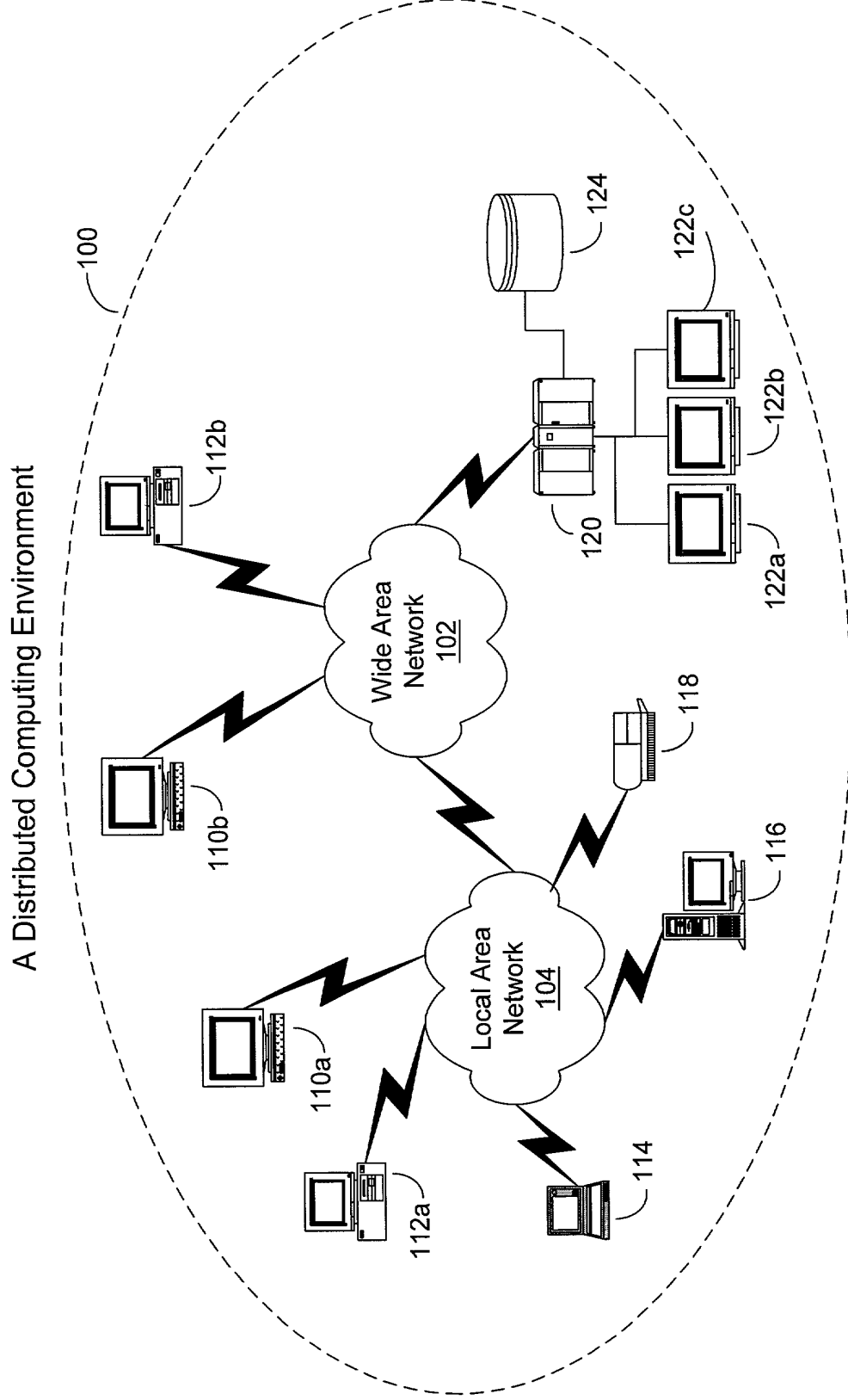


FIG. 1

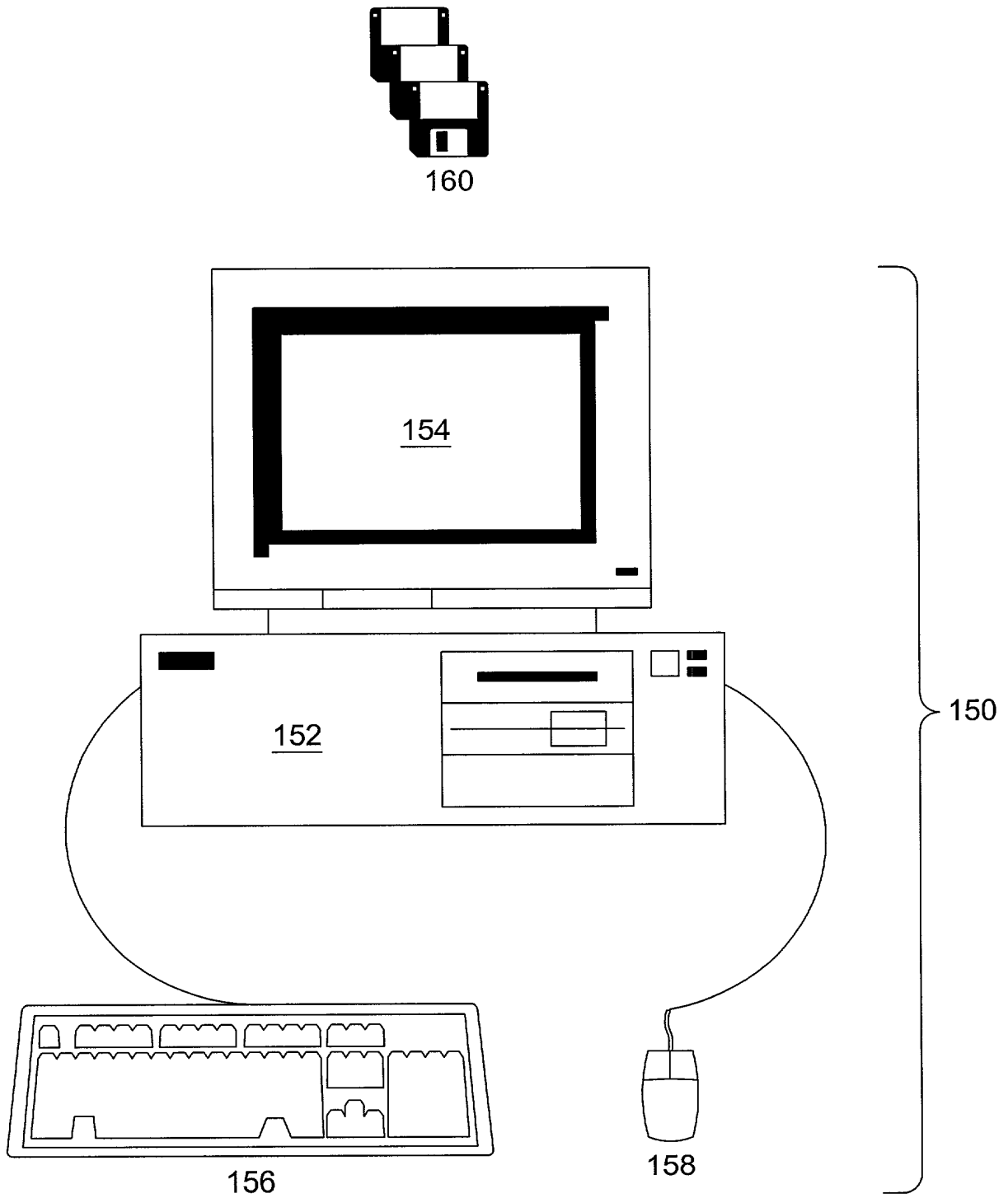


FIG. 2

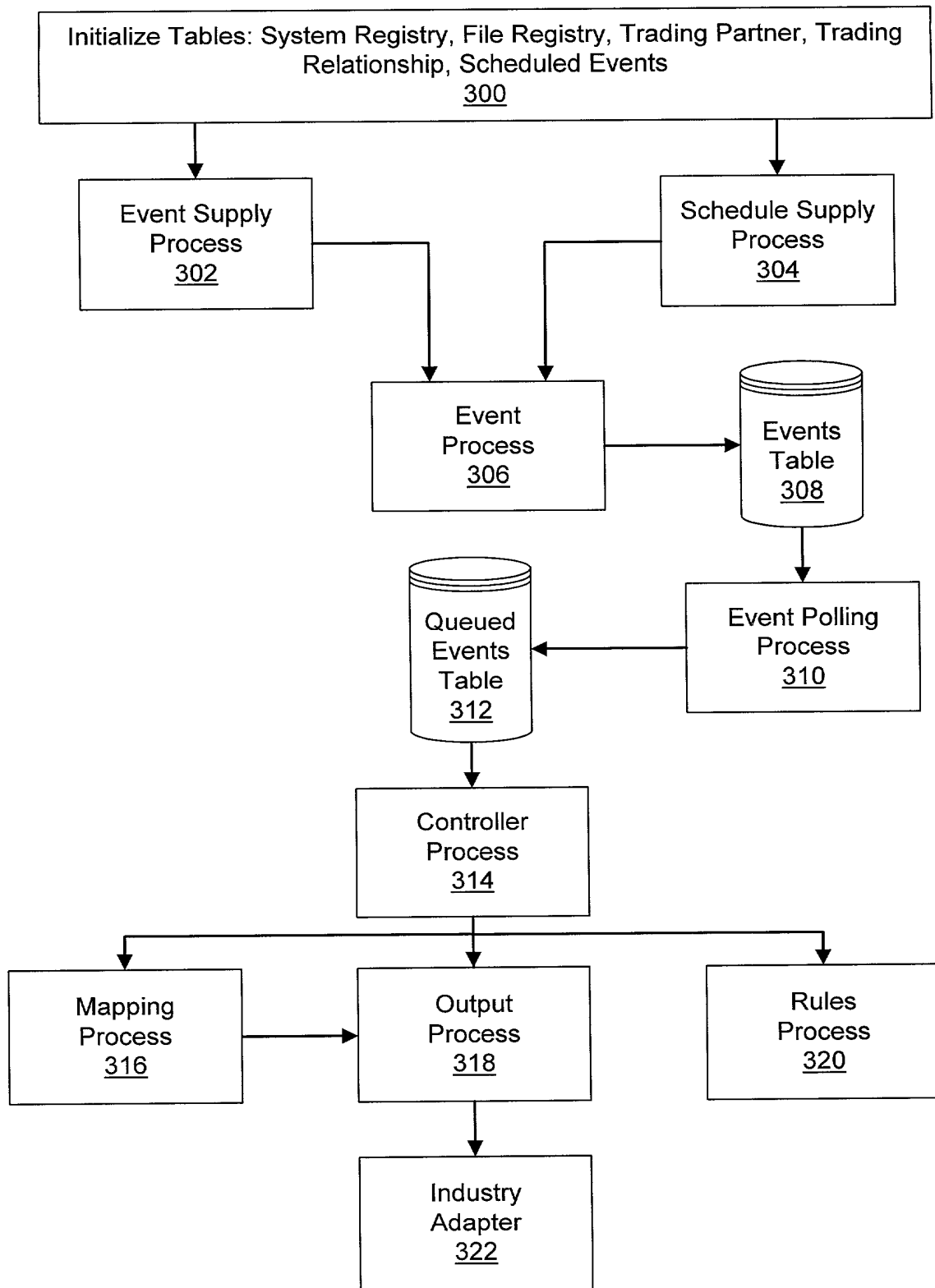


FIG. 3

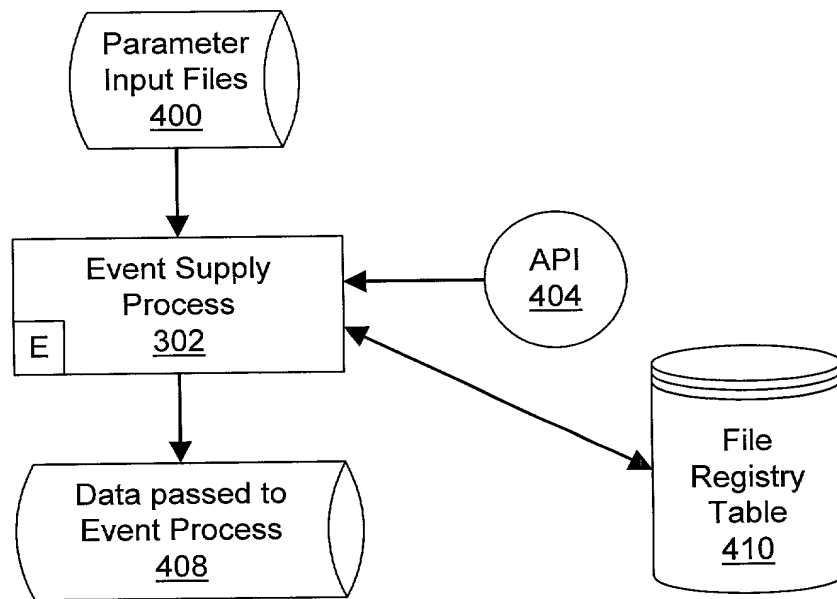


FIG. 4

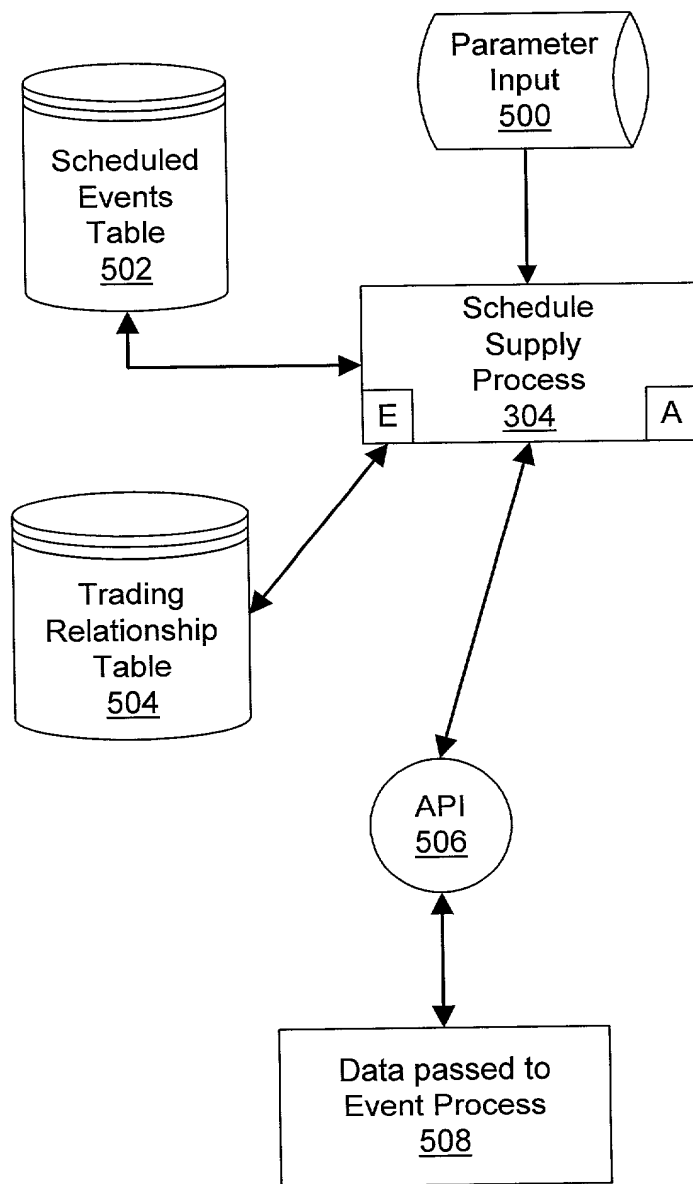


FIG. 5

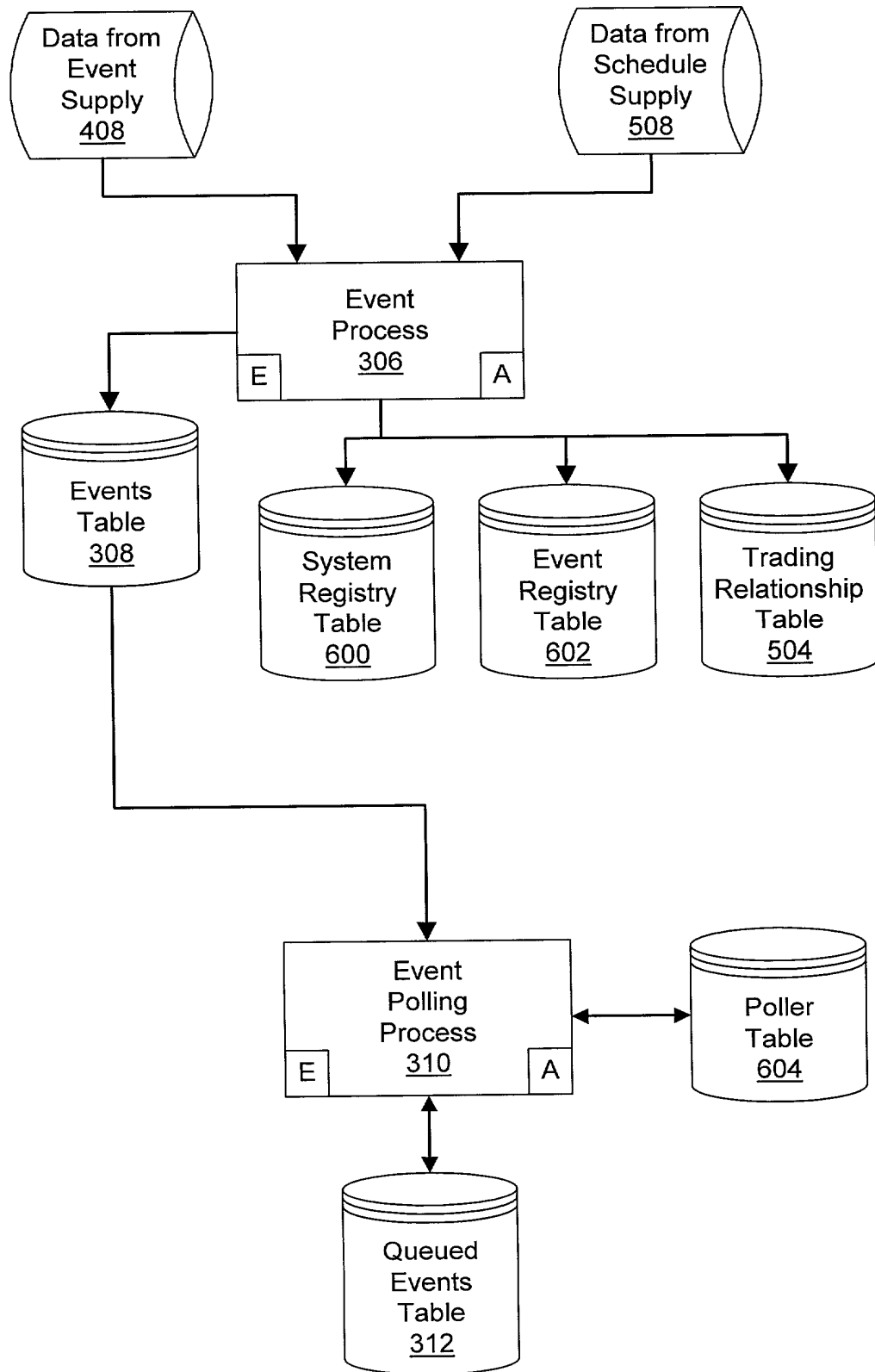


FIG. 6

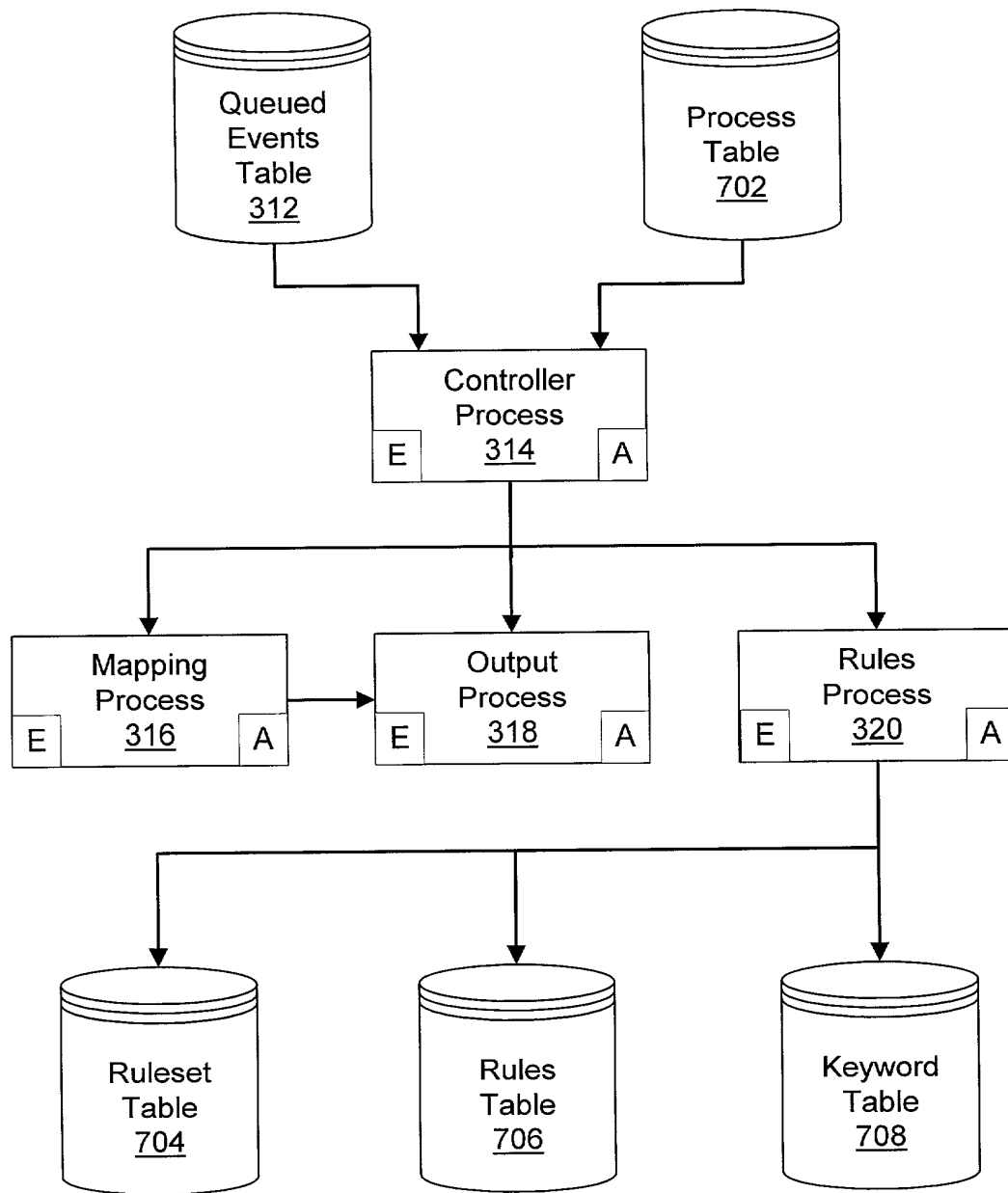


FIG. 7

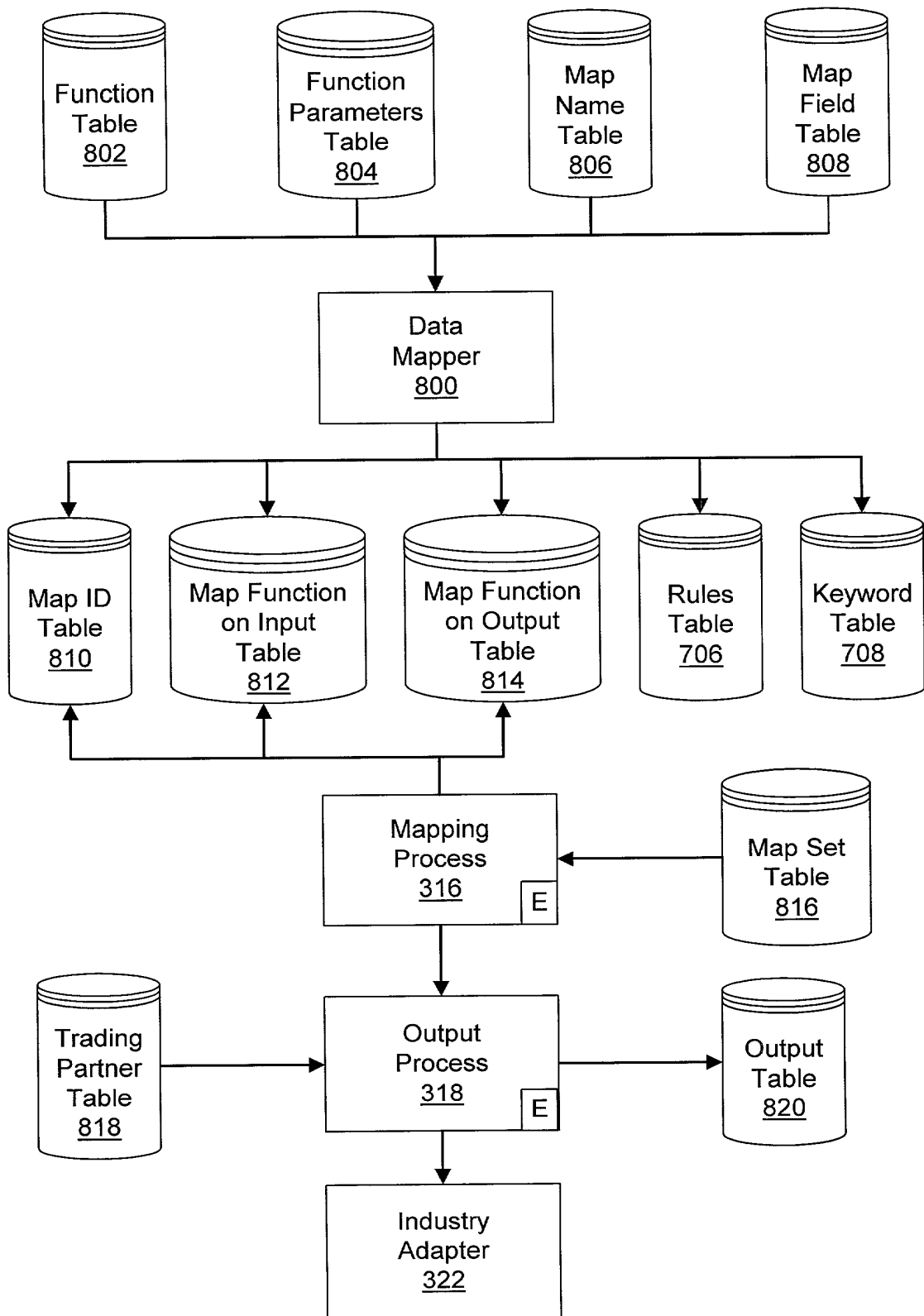


FIG. 8

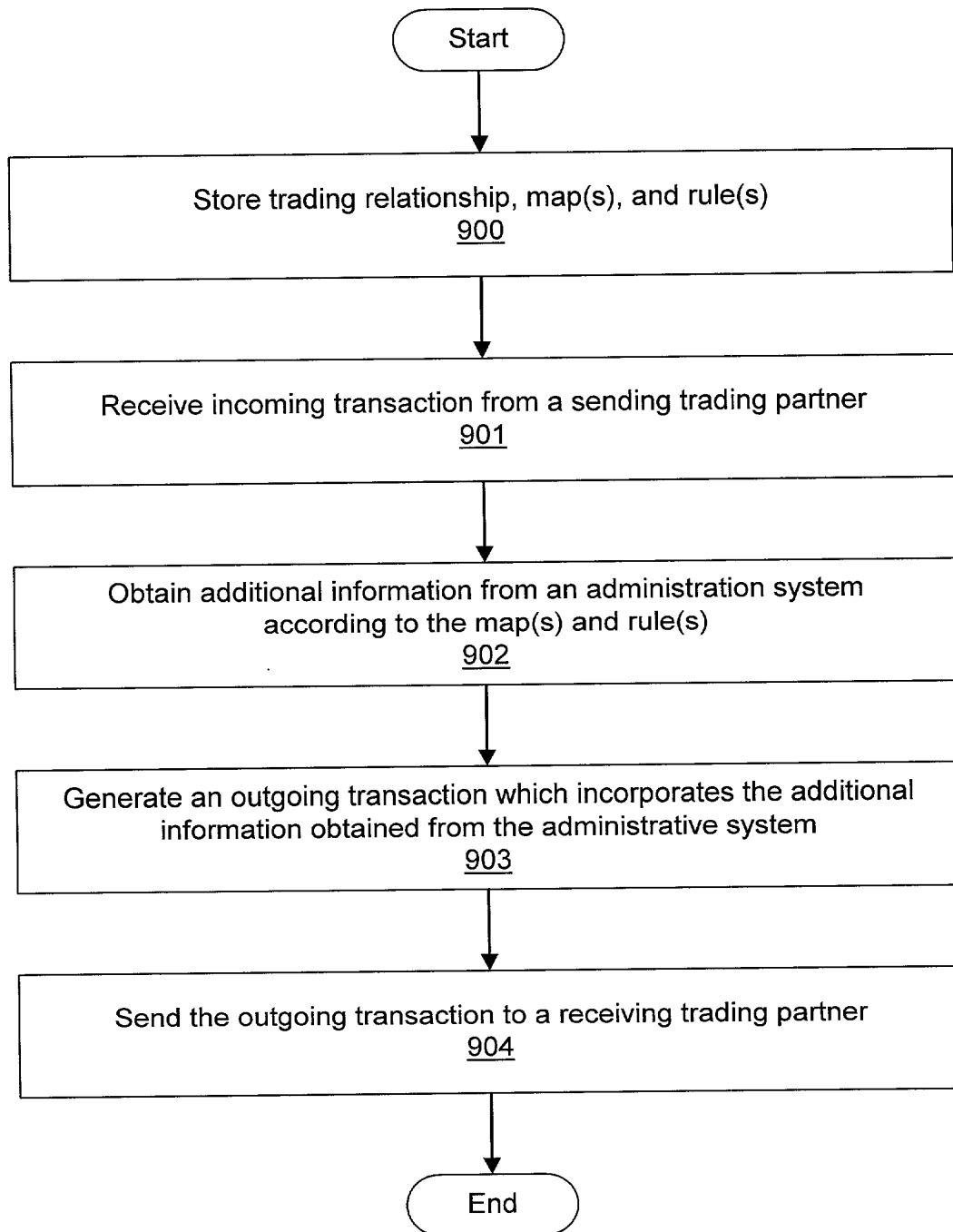


FIG. 9

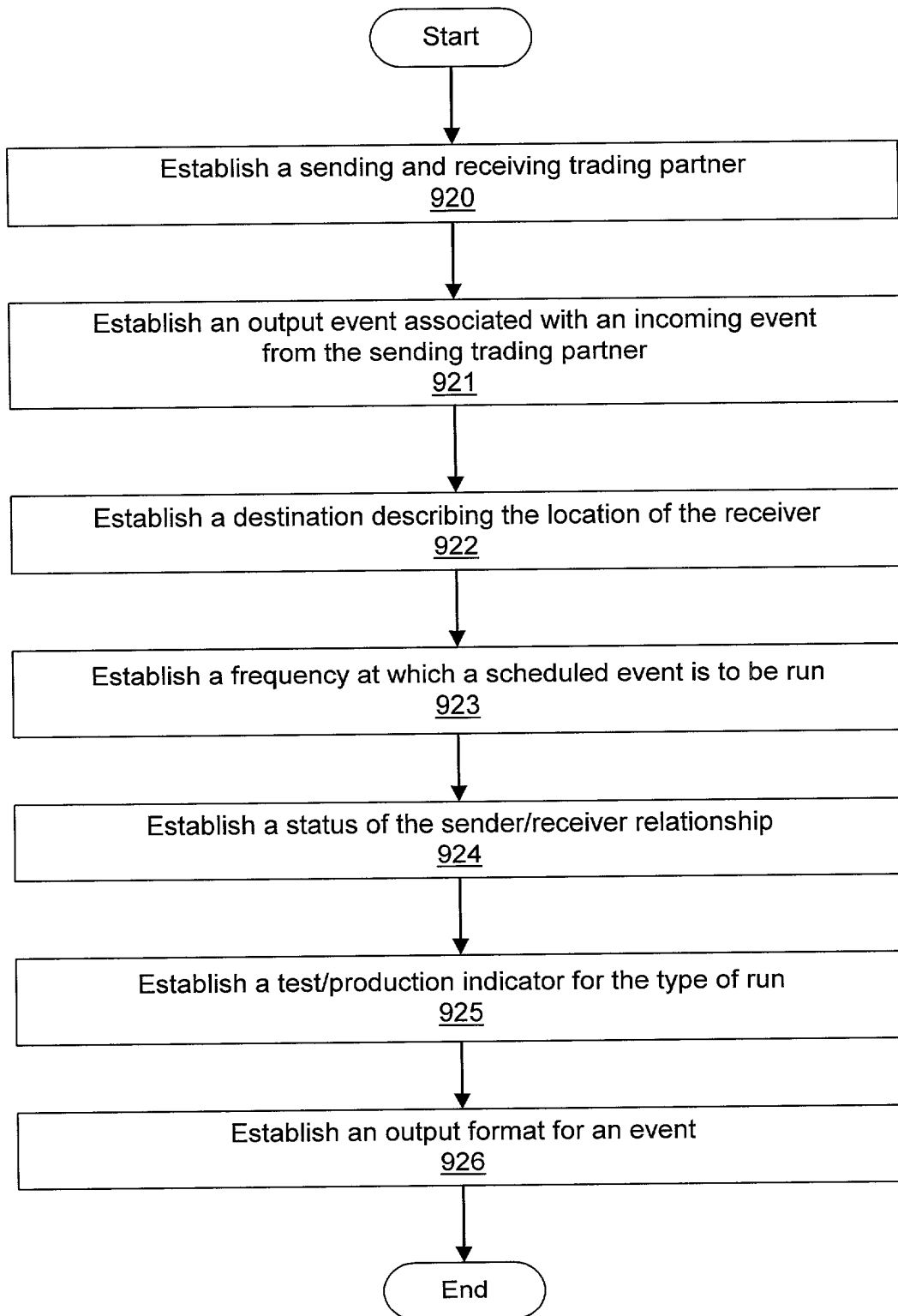


FIG. 10

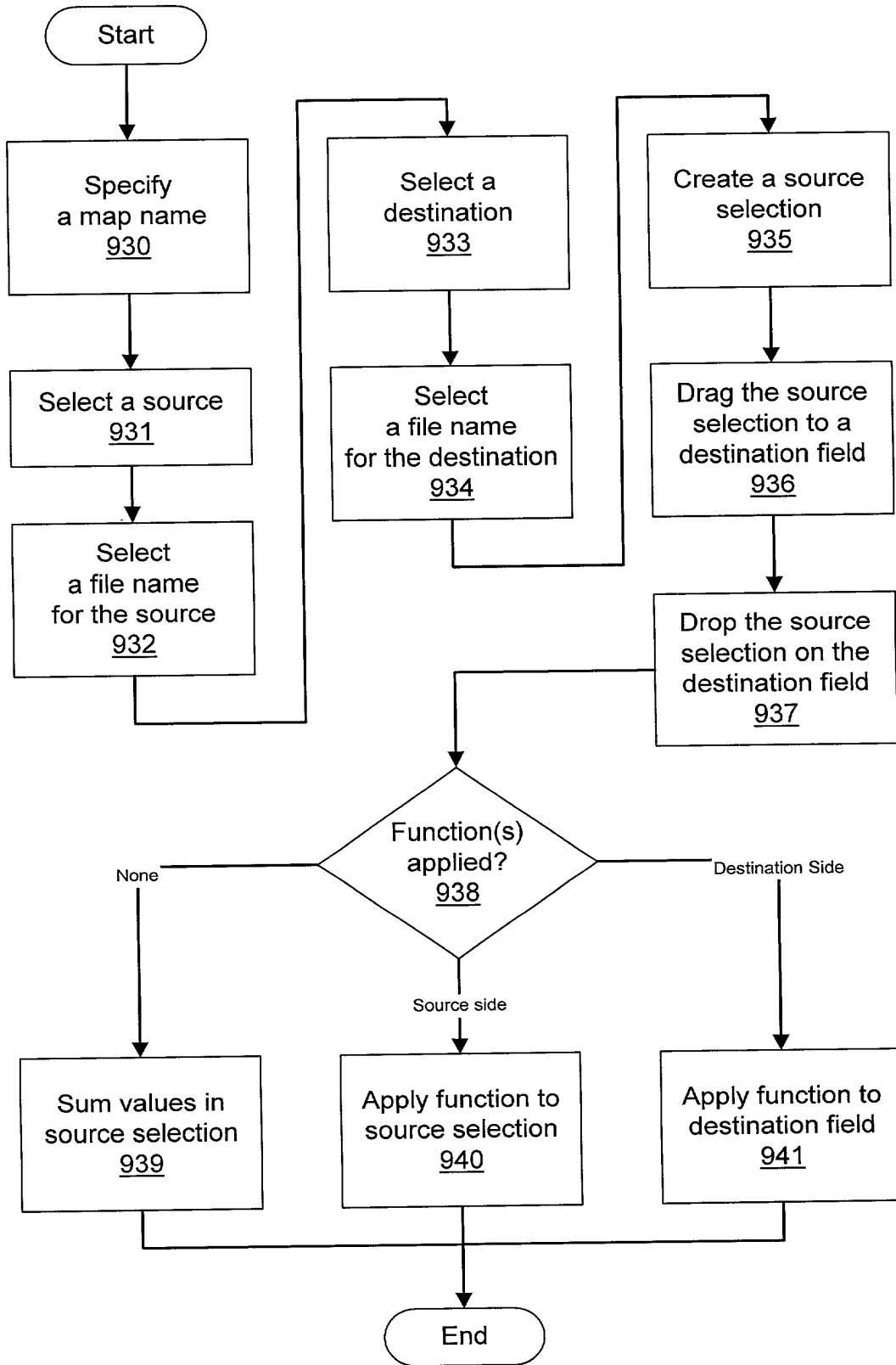


FIG. 11